Telecommun Syst (2013) 52:889–906
DOI 10.1007/s11235-011-9583-4

Reliable anycast and unicast routing: protection against attacks

Jacek Rak · Krzysztof Walkowiak

Published online: 1 September 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Recent communication networks are commonly protected against random failures, i.e. being the results of forces of nature, human errors, or hardware faults. In simulation experiments, network topologies are often assumed to be more or less regular. Known mechanisms typically refer to the case of unicast traffic protection. However, owing to the observed convergence of technologies/services, the importance of other transmission techniques (e.g. anycast, or multicast) has been increasing. Moreover, it turns out that neither failures of network elements are only the results of random faults, nor topologies of real networks are purely regular.

In this paper we introduce a novel technique, called RA (the abbreviation for "resistant-to-attacks") of protecting the anycast and unicast flows against attacks on irregular networks. In particular, we propose a new metric of link costs to be used in working path computations with the objective to avoid traversing the nodes of high degree (i.e. vulnerable to attacks). The extent of losses after attacks is further decreased by locating the anycast replica servers at low-degree nodes.

The ILP model for joint optimization of anycast and unicast flows has been formulated and followed by the time-efficient heuristic algorithm. Path protection scheme for the case of protection against a single node failure is assumed. For each anycast demand, working and backup replica servers are located at different network nodes (disjoint replica model).

Simulation results confirm that our approach provides a remarkable decrease (up to 7.47 times) in terms of the total number of connections broken due to attacks, compared to the results for the common case of locating the replica servers at high-degree nodes, and utilizing the metric of distance to find both working and backup paths.

Keywords Reliability · Protection · Anycast · Unicast · Optimization

1 Introduction

Failures of communication networks elements are inevitable. They are caused most frequently by forces of nature (e.g. hurricanes), or human errors (e.g. cable cuts). Their results may be severe. For instance, a failure of a wide-area network link offering a number (e.g. 80) of parallel transmission channels (10–40 Gbps each) implies huge data and revenue losses. As presented in [18], a failure affecting more than 30 thousand users happens once every two days, while the time to repair is up to 12 hours.

Therefore, *network survivability* being the ability to provide the continuous service in the presence of failures [2, 3, 21] is one of the most important aspects of network design. According to [12], failures of single links are most probable. Network nodes may also fail. However, as shown in [6], in high-speed wide-area networks, nodes are usually more reliable than links. Multiple failures take place, e.g. if some links are physically routed together in a duct, and a failure, being the result of a cut, affects many of them simultaneously [17, 20].

J. Rak (✉)
Department of Computer Communications,
Gdansk University of Technology, Gdansk, Poland
e-mail: jrak@pg.gda.pl

K. Walkowiak
Department of Systems and Computer Networks,
Wroclaw University of Technology, Wroclaw, Poland
e-mail: krzysztof.walkowiak@pwr.wroc.pl

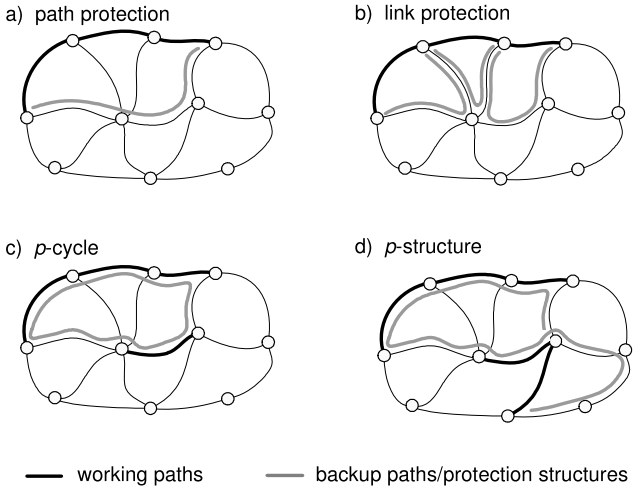


Fig. 1 Most common ways of protecting the working paths

In terms of routing, survivability is commonly assured by means of additional paths (called *backup paths*), used to provide transmission after a failure affecting the *primary* (*working*) *paths*. In order to guarantee the demanded bandwidth after the failure of a link (or a node), backup paths should not have common links (or transit nodes) with the working paths being protected, accordingly. Regarding the scope of protection, we typically choose among *path*, *segment*, or *link protection/restoration* [21], or other specialized solutions, e.g. *p*-cycles [11], or *p*-structures [16] (see Fig. 1).

Backup paths may be installed in advance (*protection scheme*), or after a failure (*dynamic restoration*). The first approach provides full recovery, but it requires a significant amount of resources (e.g. link capacity). Under dynamic restoration, there is no guarantee to find the backup paths after a failure (e.g. if link channels are not available). However, in this case, link capacities are utilized more efficiently.

Another important aspect of survivable networks design is the value of service restoration time depending linearly on the length of working and backup paths [8].

Recent papers are mainly focused on protecting the networks against random failures, i.e. being the result of hardware faults, software defects, or human errors. In this context, "random" means that the frequency of affecting various network elements (even having different topological characteristics such as node degree values) is nearly the same.

Relatively few papers address the problem of providing protection against *attacks* (i.e. malicious activities aimed to bring out severe losses at a minimum cost). This is an important issue, since a notable increase of attacks has been observed over the past few years. Unlike random failures, attacks typically affect network elements being more important than the other ones, e.g. nodes/links of relatively high degree/capacity. Therefore, they occur especially in networks having the irregular network topology (i.e. for which the degrees of nodes differ much from each other).

It is worth noting that topologies of many recent networks (e.g. Internet topology shown in Fig. 2) are not regular. As noticed by Barabási and Albert in [5], this is due to the *preferential attachment* rule frequently observed in the process of a network growth. According to this rule, it is more probable to attach a new node to an existing node n of high rather than low degree, as given in (1). It implies that nodes being already highly connected are more likely to obtain new neighbors, and as a result, their degree is increasing even faster.

$$\Pi(n) = \frac{\deg(n)}{\sum_j \deg(j)} \tag{1}$$

where $\deg(n)$ is the degree of node n defined as the number of incident links Fig. 3 illustrates an example process of a network growth. At each step, 4 new nodes are added to the network according to the preferential attachment rule (1). This example shows that the irregular topology may be obtained after adding only several new nodes.

The nodes of the highest degrees (e.g. nodes 2 and 3 in Fig. 3d), are often referred to as *central nodes* (or shortly *centers*). They are typically connected to other nodes by high-capacity links, and switch/store large amount of data. Therefore, they are excellent targets of attacks. What is more, under shortest path routing aimed at minimizing the length of the path (often met in practice), many shortest paths traverse these central nodes. As a result, after attacking any central node, many connections become affected. Shortest path routing, despite being suitable for regular networks, thus seems not to be proper for irregular networks, especially if protection against malicious activities is considered.

Previous research on reliable networks was mainly focused on *unicast* (i.e. one-to-one) communications. However, another transmission technique—*anycast* (i.e. one-to-one-of-many) has also become important, especially owing to its utilization in peer-to-peer systems (P2P), DNS service, or Content Delivery Networks (CDNs) [7, 13], and others. In anycast communications, information is replicated and stored in several replica servers located at different network nodes. A particular replica server may be chosen as the source/destination node of transmission based on several criteria, e.g. location of the client node, delay, or QoS. Regarding the protection issues, anycast backup path may lead to the same or a different replica server. The latter case (shown in Fig. 4) provides additional protection against a failure of one of the end nodes of transmission (which is not possible for unicast transmission).

The main novelty of this paper is a new approach to provide protection of anycast and unicast flows against attacks. In particular, owing to a special metric used to find working

MOST WIEDZY Downloaded from mostwiedzy.pl

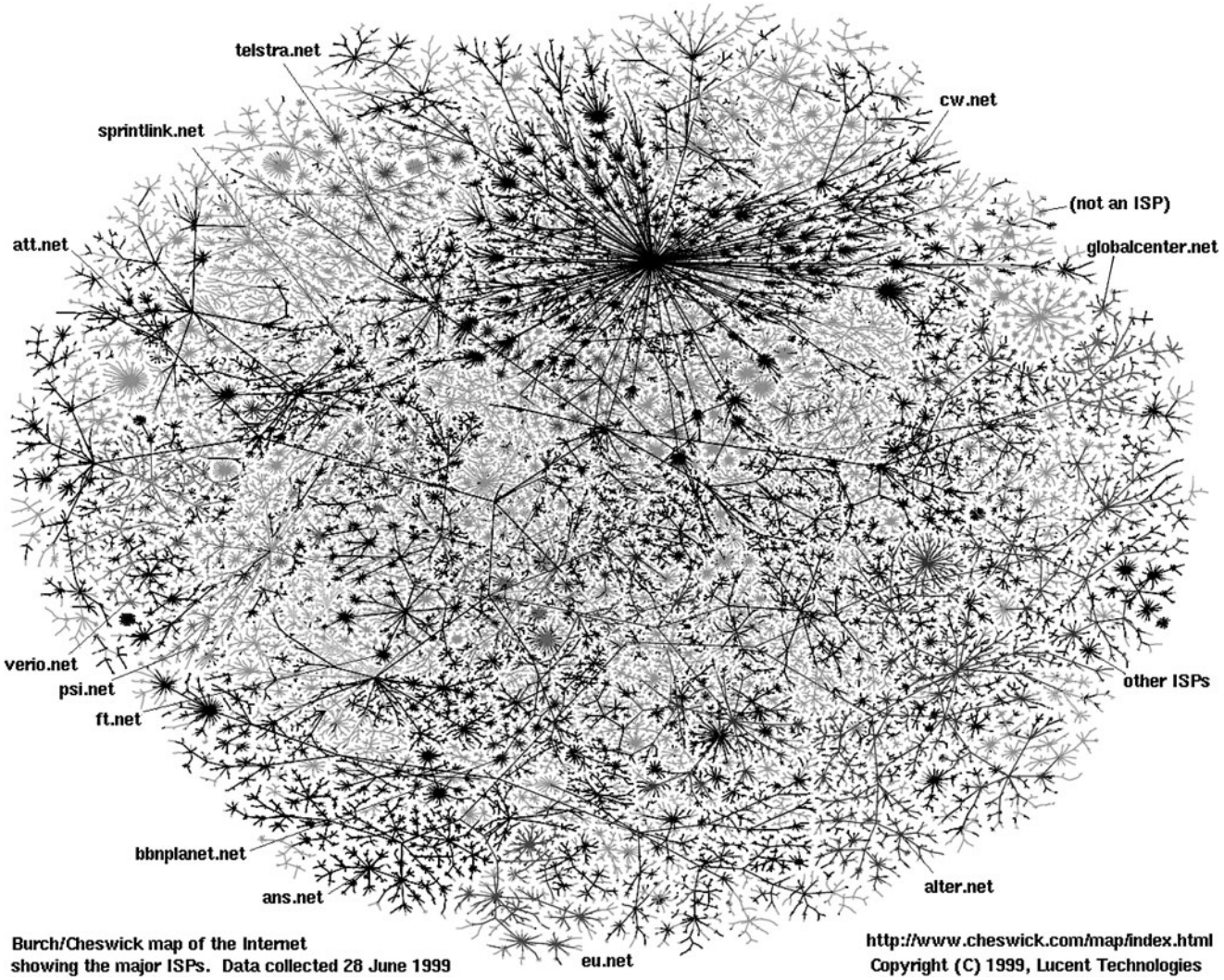


Fig. 2 Internet topology

paths (different than the standard metric of distance, which is applied here in backup path computations only), working paths omit high-degree nodes (i.e. being vulnerable to attacks). Backup paths are found as the shortest ones. They traverse central nodes, but are used only after a failure for a short time.

Additionally, replica servers are located at low-degree nodes (i.e. of low probability of breaking due to attacks). As a result, the proposed approach remarkably reduces the number of connections broken after attacks (compared to the typical case of using the distance metric in both working and backup path computations, as well as locating the replica servers at high-degree nodes).

The respective ILP formulation to find working and backup paths is presented and followed by an efficient heuristic algorithm. It is also worth noting that protection of anycast flows against attacks has not been addressed in the literature before.

The rest of the paper is organized as follows. Section 2 presents the related work. General assumptions of the proposed approach together with the ILP model are described in Sect. 3. An efficient heuristic algorithm is next introduced in Sect. 4. Section 5 presents the results of simulations performed using CPLEX 11.0 solver [15], as well as the heuristic approach.

2 Related work

In anycast (one-to-one-of-many) communications, information is replicated and stored in several replica servers typically located at different nodes. This is a commonly accepted technique frequently utilized by caching and replication systems, including Peer-to-Peer (P2P) systems, overlay networks, wireless sensor networks, web service, distributed database systems, etc. An excellent example of anycast com-

MOST WIEDZY Downloaded from mostwiedzy.pl

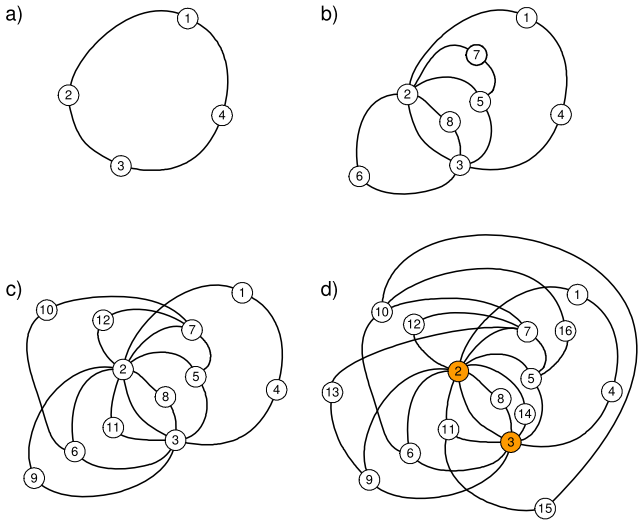


Fig. 3 Example evolution of a network topology based on the preferential attachment rule

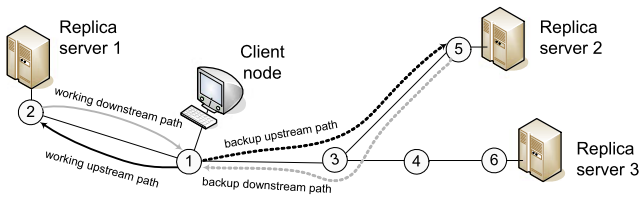


Fig. 4 Example anycast connection (primary and backup replica servers are located here at different nodes)

munications is the Content Delivery Network (CDN), which is used to deliver the requested content to end users on behalf of origin Web servers. In CDNs, the original content is offloaded from the source server to other servers located at different nodes. The most popular CDN system widely used in the Internet by larger web sites is Akamai [7, 13].

Previous works focus mainly on applications of anycast transmission in IP networks using connectionless transmission [4, 13]. Anycasing in connection-oriented networks is considered less frequently. In [14], a generalized RWA (Routing and Wavelength Assignment) problem is introduced, where lightpaths can be not only unicast but also anycast or multicast connections. The objective there is to establish a given set of connections while minimizing the number of wavelength channels.

Another interesting approach to provide protection against failures for anycast and unicast transmission is presented in [22]. In particular, the authors propose a model of joint optimization of anycast and unicast flows. Restoration methods are applied there to find the backup paths for broken connections.

All the works mentioned above refer to the case of protection against random failures. Regarding the issues of anycast flows protection against attacks, there are practically no techniques available in the literature.

3 Proposed approach

The objective of the paper is to propose the survivable routing of anycast and unicast demands aimed at reducing the number of flows affected after attacks. Since the extent of losses after attacks strongly depends on the topological characteristics of the network (in particular on the distribution of node degrees), one of the possible ways to decrease it is to modify the routing (e.g. the metric of link costs) according to these characteristics.

Communication networks typically grow over time. Contrary to beliefs of many people, the initial characteristics of many networks (i.e. designed in the beginning) are rarely preserved. For instance, it often turns out that networks are evolving according to the preferential attachment rule, implying that new nodes are being attached to the already highly connected ones (see Fig. 3). In particular, the authors of [5] have shown that the growth of a network, if uncontrolled, leads to the power law distribution of node degrees ($P(k) \sim k^{-\gamma}$, where k is the degree of a node). Such power-law networks are often referred to as scale-free (SF) networks. Examples of SF networks include the Internet (with $\gamma = 2.22$ [23]), or networks shown in Fig. 13.

As stated earlier, power law distribution of node degrees implies the existence of nodes of extraordinary high degree (being much greater than the average node degree in the network), that are the main targets of attacks. It is also easy to notice that the vulnerability of the shortest path routing to attacks strongly depends on the topological characteristics of the network. Therefore, it changes as the network topology is evolving over time.

The main proposal of the paper is as follows. In order to reduce the number of connections broken due to attacks, we introduce a new metric of link costs to be used in working path computations to make these paths omit central nodes, as shown in Fig. 5 for the case of Italian Network [1].[1] This metric is dynamic, i.e. it returns different values of link costs in respond to the changes in the network topology. Additionally, in order to decrease the number of anycast connections broken after attacks on network nodes, we propose to locate the replica servers at low-degree nodes (i.e. of low probability of breaking due to attacks).

We consider here a directed network $\Gamma(N, A)$, where: N and A denote the sets nodes and directed arcs, accordingly. Each link is thus modeled by a pair of unidirectional arcs: $a_h = (i, j)$ and $a_{h'} = (j, i)$.

[1] Topology of Italian network is neither scale-free, nor regular. However, its node degree distribution is not uniform, and nodes 6, 11, and 17 may be referred to as central nodes.

MOST WIEDZY Downloaded from mostwiedzy.pl

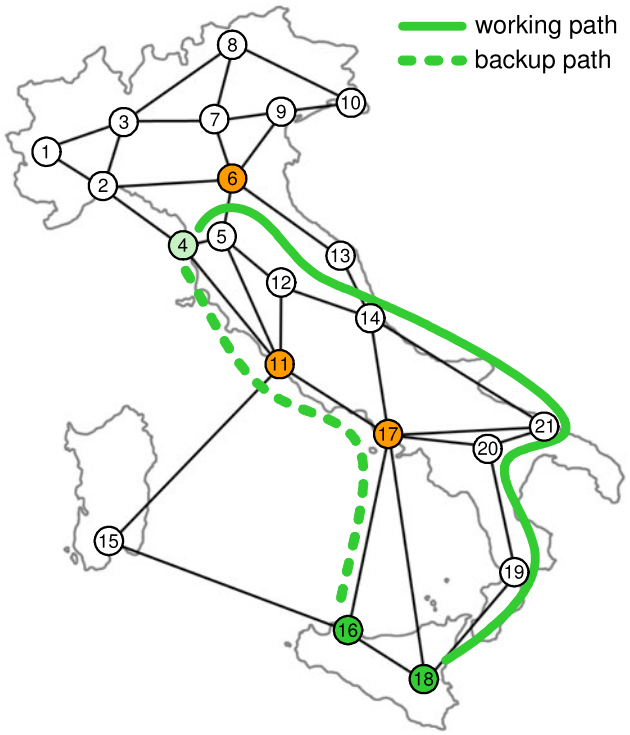


Fig. 5 Example anycast routing according to the proposed approach

Each arc ah $a_h \in A$:

– is characterized by its length s_h and a pair of costs: ζ_h and ξ_h to be used in working and backup path computations, respectively,
– offers Λ unidirectional channels of equal capacity.

In this paper, network flows are modeled as the non-bifurcated multicommodity flows. Anycast and unicast demands are represented by pairs of unidirectional demands (in opposite directions).

For the purpose of working path computations, we propose to use the metric based on the so called *betweenness centrality* parameter (BC) defined in [10] and shown in (2), since it provides us with good estimates on the centrality of node n, and therefore on its vulnerability to attacks.

$$BC(n) = \sum_{p \neq q} \frac{\#sp_n(p,q)}{\#sp(p,q)} \tag{2}$$

where:

$\#sp(p,q)$ is the number of the shortest paths between nodes p and q of the same minimal length
$\#sp_n(p,q)$ is the number of the shortest paths between nodes p and q of the same minimal length, traversing node n

In this paper, the cost ζ_h of any arc a_h used in working path computations is defined in (3) as the average value of the normalized betweenness centrality parameter (BC^*) of nodes n_i and n_j being incident to the arc a_h. Since each central node has a large value of betweenness centrality parameter, then the cost of any link incident to a central node is high as well. Working paths will thus omit such network elements, which in turn will make them resistant-to-attacks.

$$\zeta_h = \zeta_{(i,j)} = \frac{BC^*(i) + BC^*(j)}{2} \tag{3}$$

where

$$BC^*(n) = \frac{BC(n)}{\max_i (BC(i))} \tag{4}$$

In the case of backup path computations, the cost ξ_h of any arc a_h is defined in 5 as the normalized length of this arc. Therefore, backup paths are found here as the shortest ones.

$$\xi_h = \frac{s_h}{\max_i (s_i)} \tag{5}$$

where s_h is the length of arc a_h.

In the remaining part of the paper, we call our approach the RA (the abbreviation for "resistant-to-attack") method, in contrast to the common technique (here referred to as NA—i.e., "non-resistant-to-attack") of finding both working and backup paths as the cheapest ones in terms of path length (using (5) to find working and backup paths), as well as locating the replica servers at high-degree nodes.

Path protection scheme with protection against a single node failure is assumed. It means that for each working path, there is one backup path (having no common transit nodes with its working path) found in advance.

In simulations we assume that the capacity f_r requested for each demand d_r is equal to the capacity of a single link channel. The capacity of each link channel is unitary. Therefore, for each demand d_r we have $f_r = 1$.

In the ILP model introduced below, for anycast demands we use indices $r = 1 \ldots |D_{AN}|$, while unicast demands are described by indices $r = |D_{AN}| + 1 \ldots |D|$, where $|D|$ is the total number of demands.

Indices

$\Gamma(N, A)$ directed network; N and A denote the sets of nodes and directed arcs, accordingly
$n(h)$ index of a node (an arc)
D^{UN} set of unicast demands
D^{AN} set of anycast demands. Two types of anycast demands may be distinguished: downstream or upstream. Each anycast demand is defined by the client node (i.e. the source node for the upstream demand and the destination node for the downstream demand). The respective upstream and downstream demands having the same client node are associated

MOST WIEDZY Downloaded from mostwiedzy.pl

D^{US} set of anycast upstream demands (i.e. from the client node to the replica server)
D^{DS} set of anycast downstream demands (i.e. from the replica server to the client node)
D set of all demands ($D = D^{UN} \cup D^{AN}$)
R set of nodes that host the replica servers
r index of a demand (anycast or unicast)
$\tau(r)$ index of anycast demand associated with demand r
$s_r(t_r)$ source (destination) node of the r-th demand. For upstream (downstream) anycast demands, only the source (destination) node $s_r(t_r)$ is given, accordingly

Variables

$x_{r,h}(y_{r,h})$ equals 1, if the channel of an arc $a_h = (i, j)$ is allocated for a working (backup) path of the r-th demand, accordingly; 0 otherwise
$z_{r,n}(v_{r,n})$ equals 1, if replica node n is selected as a working (backup) replica of the r-th anycast demand, accordingly; 0 otherwise

Objective

It is to find working and backup paths transporting the required flows from sources to destinations protected against a single node failure by single backup paths and minimizing the following linear cost:

$$\varphi(\mathbf{x}) = \sum_{r \in D} \sum_{h \in A} \left(\zeta_h x_{r,h} + \xi_h y_{r,h}\right) \tag{6}$$

where $\zeta_h(\xi_h)$ is the cost per unit flow of each commodity on the arc a_h of a working (backup) path, accordingly

Constraints

(a) providing flow conservation for the working paths of unicast demands[2]:

$$\sum_{\substack{h \in \{h: a_h \equiv (n,j) \in A;\\ j \in N; j \neq n\}}} x_{r,h} - \sum_{\substack{h \in \{h: a_h \equiv (i,n) \in A;\\ i \in N; i \neq n\}}} x_{r,h} = \begin{cases} 1 & \text{if } n = s_r \\ -1 & \text{if } n = t_r \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

for each $r \in D^{UN}$ and $n \in N$;
$a_h = (i, n)$ = arc incident into a node n;
$a_h = (n, j)$ = arc incident out of a node n

(b) providing flow conservation for the working paths of anycast downstream demands[3]:

$$\sum_{\substack{h \in \{h: a_h \equiv (n,j) \in A;\\ j \in N; j \neq n\}}} x_{r,h} - \sum_{\substack{h \in \{h: a_h \equiv (i,n) \in A;\\ i \in N; i \neq n\}}} x_{r,h} = \begin{cases} z_{r,n} & \text{if } n \in R \\ -1 & \text{if } n = t_r \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

for each $r \in D^{DS}$ and $n \in N$

(c) providing flow conservation for the working paths of anycast upstream demands[4]:

$$\sum_{\substack{h \in \{h: a_h \equiv (n,j) \in A;\\ j \in N; j \neq n\}}} x_{r,h} - \sum_{\substack{h \in \{h: a_h \equiv (i,n) \in A;\\ i \in N; i \neq n\}}} x_{r,h} = \begin{cases} 1 & \text{if } n = s_r \\ -z_{r,n} & \text{if } n \in R \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

for each $r \in D^{US}$ and $n \in N$

(d) for working paths of associated anycast upstream and downstream demands d_r and $d_{\tau(r)}$ (d_r and $d_{\tau(r)}$ must use the same working replica node)[5]:

$$z_{r,n} = z_{\tau(r),n} \tag{10}$$

for each $r \in D^{DS}$ and $n \in R$

(e) or working paths of anycast demands (each anycast demand must be assigned to exactly one working replica node[6]):

$$\sum_{n \in R} z_{r,n} = 1 \tag{11}$$

for each $r \in D^{AN}$

(f) on finite arc capacity:

$$\sum_{r \in D} (x_{r,h} + y_{r,h}) \leq \Lambda \tag{12}$$

for each $h \in A$

[2] In order to provide the respective constraints for backup paths of unicast demands, in (7) $x_{r,h}$ must be replaced by $y_{r,h}$.

[3] The respective constraints for backup paths of anycast downstream demands may be obtained from (8) by replacing $x_{r,h}$ and $z_{r,h}$ by $y_{r,h}$ and $v_{r,h}$, accordingly.

[4] Equations for backup paths of anycast upstream demands are similar to (9) with $x_{r,h}$ and $z_{r,h}$ replaced by $y_{r,h}$ and $v_{r,h}$, accordingly.

[5] Constraints for backup paths of associated anycast upstream and downstream demands d_r and $d_{\tau(r)}$ are similar to (10) with $z_{r,n}$ and $z_{\tau(r),n}$ replaced by $v_{r,n}$ and $v_{\tau(r),n}$, accordingly.

[6] In order to provide the respective backup replica node constraints for backup paths of anycast demands, in (11) $z_{r,n}$ should be replaced by $v_{r,n}$.

MOST WIEDZY Downloaded from mostwiedzy.pl

(g) to ensure that every backup path is node-disjoint with its working path:

$$\sum_{\substack{h\in\{h:a_h\equiv(n,j)\in A;\\ j\in N;\, j\neq n\}}} (x_{r,h}+y_{r,h}) \le 1 \tag{13}$$

$$\sum_{\substack{h\in\{h:a_h\equiv(i,n)\in A;\\ i=N;\, i\neq n\}}} (x_{r,h}+y_{r,h}) \le 1 \tag{14}$$

for each $r \in D$ where $n \neq s_r$; $n \neq t_r$ for unicast demand $r \in D^{UN}$ for transit nodes (if both paths consist of at least two arcs); $n \neq s_r$; $n \notin R$ for upstream anycast demand $r \in D^{US}$ for transit nodes (if both paths consist of at least two arcs); $n \neq t_r$; $n \notin R$ for downstream anycast demand $r \in D^{DS}$ for transit nodes (if both paths consist of at least two arcs)

(h) providing nonnegativity of variable values:

$$x_{r,h} \ge 0; \qquad y_{r,h} \ge 0; \qquad z_{r,n} \ge 0; \qquad v_{r,n} \ge 0 \tag{15}$$

for each $r \in D$, $h \in A$ and $n \in N$

(i) providing the nodal disjointedness of working and backup replica nodes for anycast demands:

$$\sum_{n\in R}(z_{r,n}+v_{r,n}) \le 1 \tag{16}$$

for each $r \in D^{AN}$

Formulas (6)–(15) define a model of survivable routing that provides protection of anycast and unicast demands against a single node failure. Constraint (16) is to provide additional protection against the working replica node failure (i.e. to ensure that for each anycast demand, working and backup replica servers are located at different nodes).

The problem defined by formulas (6)–(16) is NP-complete since even the much the simpler task to find $|D|$ working paths only (without protection) in a capacity-constrained network is NP-complete [18]. Therefore, apart from this model, we introduce in the next section a time-efficient heuristic algorithm to find the solutions for larger problem instances.

However, as stated in [19], in the case of multi-cost networks (i.e. when for any link, different link costs may be used to find working and backup paths), e.g. in the issue considered here, the problem is NP-complete even for a single demand. Therefore, any heuristic algorithm is able to return only the suboptimal results even for the simplest case of finding the paths for one demand only.

4 Heuristic algorithm

In this section we present our heuristic algorithm to calculate disjoint paths for the scenario of differentiated link costs (i.e. the case where for each arc a_h, the costs ζ_h and ξ_h to be used in working and backup path computations, accordingly, may be different). The algorithm presented in Fig. 6 is an extension to the k-Penalty approach introduced by the authors in [19], originally designed for the case of protecting the unicast traffic against a simultaneous failure of $k-1$ network elements. In this paper, the algorithm is extended to provide protection for anycast demands (in particular under assumption that working and backup replica servers are located at different network nodes).

In this paper we analyze the properties of the approach for the case of a path protection scheme applied to provide protection against a failure of a single node. Therefore, the algorithm is used here to find $k = 2$ node-disjoint paths for each demand. In particular, it implies that in Fig. 6, the costs ξ_h^1 and ξ_h^2 of network arcs a_h (stored in matrices Ξ^1 and Ξ^2, respectively) are initially assigned the values of ζ_h and ξ_h, accordingly.

Similar to the *active path first approach*, for each demand the algorithm first finds the working path (using any algorithm of finding the shortest path, e.g. the Dijkstra's algorithm [9]). The difference is that, in our algorithm when finding the backup path, the costs of the respective *forbidden arcs* used by the working path, instead of being set to infinity (in order to provide the disjointedness of the connection paths), are increased by a certain large value. This modification is introduced to avoid entering into the *trap problem* (i.e., the case when the algorithm fails to find the next path of a demand, even though the requested number of disjoint paths exists). In particular, in the considered case of $k = 2$ disjoint paths, before finding the next path η_2 (i.e. the backup path), the costs of forbidden arcs a_h to be used in backup path computations are increased by the total cost of the working path of a demand (Step 4), using the matrix of backup link costs Ξ^2.

In order to avoid entering into the trap problem, the algorithm thus allows the backup path to temporarily traverse the forbidden arcs. However, such situation implies a conflict, and, if occurred, then the respective costs ξ_h^1, ξ_h^2 of the so called conflicting arcs are permanently increased in Ξ^1, Ξ^2 by the total cost of the recently found path η_2 in terms of the link costs from Ξ^1 and Ξ^2, accordingly. In this case, the execution of the algorithm also starts from the beginning (see Step 6.3 in Fig. 6).

It is worth mentioning that after several possible conflicts, the algorithm almost always terminates successfully. In particular, during experiments, the probability of exceeding the upper bound *it_upper* on the number of allowed conflicts was each time less than 0.034.

The time complexity of the algorithm depends on the algorithm used to find the shortest paths in Step 5. In the case the Dijkstra's algorithm [9] is used for this purpose, the complexity of the algorithm is bounded from above by $O(|N|^2)$,

MOST WIEDZY Downloaded from mostwiedzy.pl

INPUT
- A demand d_r to find the set of $k = 2$ node-disjoint paths (unicast demand is determined by a pair of nodes (s_r, t_r), while anycast demand by a given client node s_r to be connected to replica servers located at different network nodes)
- The matrices Ξ^1, Ξ^2 of arc costs ξ_h^1 and ξ_h^2 (for working and backup path computations, accordingly)
- The upper bound *it_upper* on the number of allowed conflicts

OUTPUT The set of $k = 2$ node-disjoint paths η_1, η_2
VARIABLES Ξ^{tmp} auxiliary matrix of arc costs ξ_h^{tmp}
j index of the path currently to be found
ic conflict counter

Step 1 Set $ic = 1$.
Step 2 Set $j = 1$.
Step 3 For each network arc a_h, set $\xi_h^{tmp} = \xi_h^j$.
Step 4 For each path η_i from the set of previously found $j - 1$ paths and for each arc a_h, if a_h is a *forbidden arc** of the path η_i, then increase the arc cost ξ_h^{tmp} by the total cost ξ^i of η_i in Ξ^j.
Step 5 Find η_j using the Dijkstra's algorithm and the costs matrix Ξ^{tmp}.
Step 6 If η_j is not disjoint with the previously found $j - 1$ paths then:
6.1 Increase the costs ξ_h^i of each *conflicting arc*** a_h of η_j by the total cost ξ^j of η_j in all matrices Ξ^i, accordingly. After that, delete the found paths.
6.2 Set $ic = ic + 1$.
6.3 if $ic > it_upper$ then
terminate and reject the demand,
else go to Step 2.
else set $j = j + 1$.
Step 7 If $j > 2$ then terminate and return the found set of paths
else go to Step 3.

* A forbidden arc of η_i is an arc that is traversed by η_i, or is incident to any transit node of η_i (for link and nodal disjointedness, respectively)
** An arc a_h is a conflicting arc of a given path η_j, if:
– is jointly used by η_j and by any other of previous $j - 1$ paths (when link disjointedness is required), or
– is incident to any common transit node jointly used by η_j and by any other of previous $j - 1$ paths (in the case of nodal disjointedness)

Fig. 6 The proposed algorithm to find $k = 2$ node-disjoint paths of a demand (unicast or anycast)

where $|N|$ is the number of nodes. Our algorithm can be easily modified to return a set of k-disjoint paths for each demand, where k is any arbitrary positive value.

Example execution of the algorithm will be presented here for the case of Italian Network from Fig. 5. Initial costs of working and backup path links are as given in Figs. 7, 8. They are based on (3) and (5), accordingly.

Figures 9, 10, 11, 12 present the example execution of our algorithm used here to find the working and backup paths for a unicast connection between nodes 4 and 21. After finding the working path η_1: 4-11-12-14-21 (Fig. 9) the costs of the respective forbidden links to be used in backup path calculations are increased by the total cost of η_1 using the costs from Ξ^2 depicted in Fig. 8 (i.e. by 1.89). After that, the backup path η_2: 4-5-11-17-21 is found (Fig. 10). However, it has a common transit node 11 with the working path. Therefore, the costs ξ_h^1 and ξ_h^2 of all conflicting arcs a_h (i.e. arcs incident to node 11) are permanently increased by the total cost of the path η_2 calculated using the initial matrices Ξ^1 and Ξ^2 of working and backup path link costs (i.e. by 3.03 and 1.58 in Ξ^1, Ξ^2 accordingly), and computations start from the beginning. After that, the algorithm manages to find the connection paths, which are finally shown in Fig. 12. Working path η_1^*: 4-5-12-14-21 omits nodes of high-degree (and is therefore resistant-to-attacks). Backup path η_2^*: 4-11-17-20-21 is found as the shortest one. It traverses central nodes 11 and 17, but is used only after a failure for a short time.

5 Simulation results

Simulations were performed to evaluate the characteristics of the proposed approach regarding the aggregate number of connections broken due to attacks (i.e. the total number of broken connections in a single simulation), link capacity utilization ratio, the length working and backup paths, as well as the time of connection restoration. Time of connection restoration was calculated according to [18]. Numerical experiments were performed using CPLEX 11.0 and the respective heuristic algorithm for the topologies of two irregular networks shown in Fig. 13, i.e. the ASF Network and BA-150 Network, consisting of 15 and 150 nodes, accordingly. These topologies were obtained using the Barabási-Albert algorithm of topology generation from [5].

For each anycast and unicast demand d_r, we assumed the following properties:

– the demanded capacity equal to the link channel capacity (i.e. unitary),

MOST WIEDZY Downloaded from mostwiedzy.pl

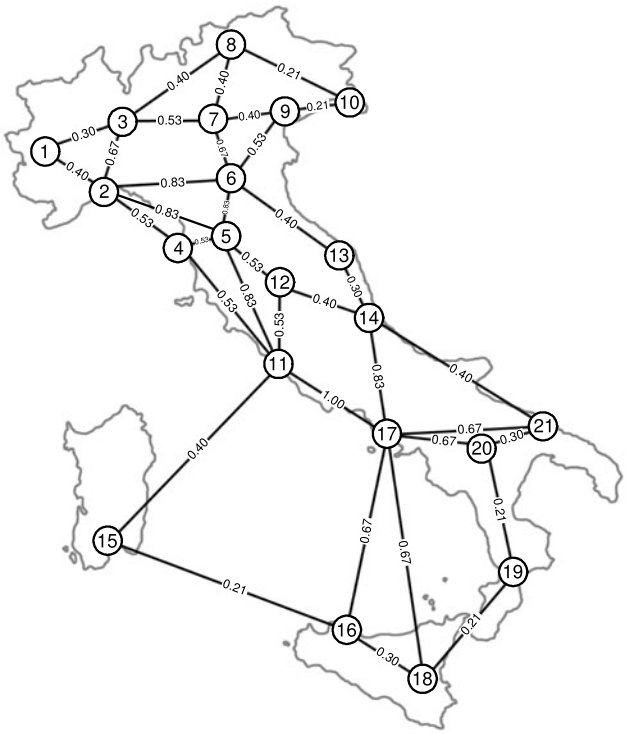


Fig. 7 Initial costs of links used in working path computations calculated for the example Italian network

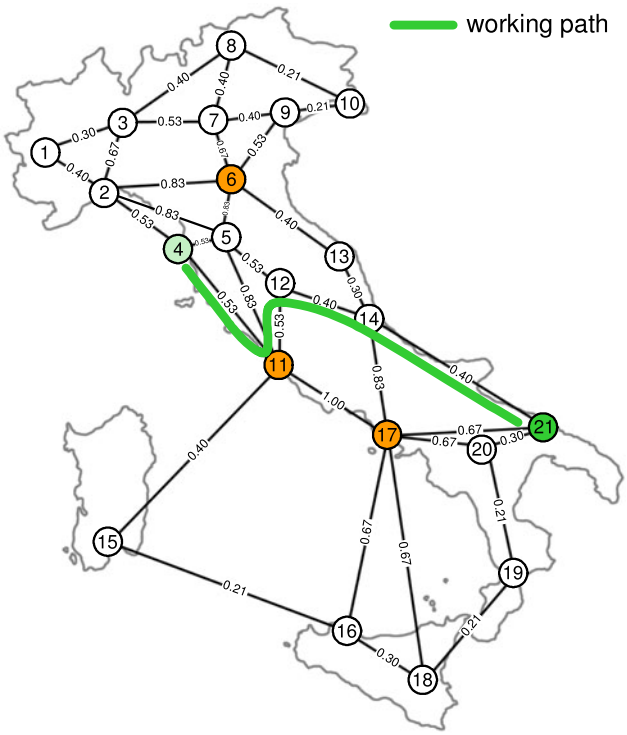


Fig. 9 Round 1 of the example execution of the proposed algorithm

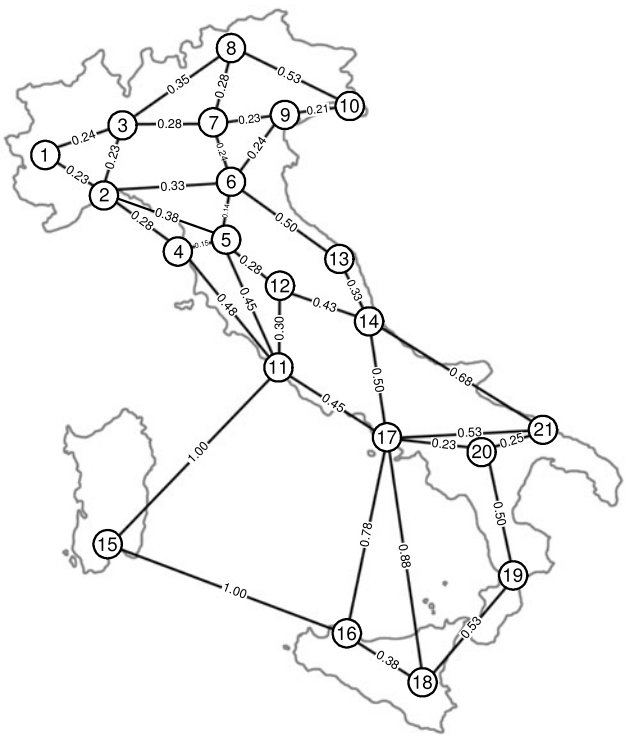


Fig. 8 Initial costs of links used in backup path computations calculated for the example Italian network

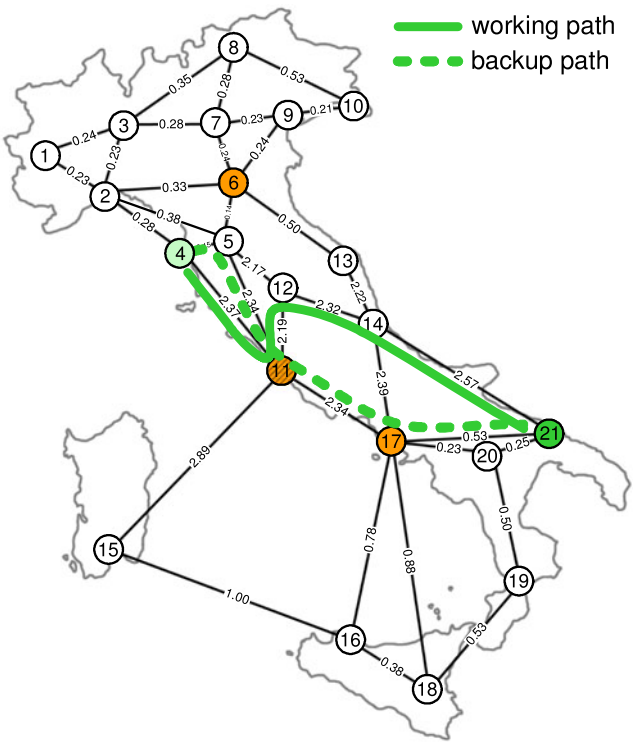


Fig. 10 Round 2 of the example execution of the proposed heuristic algorithm

MOST WIEDZY Downloaded from mostwiedzy.pl

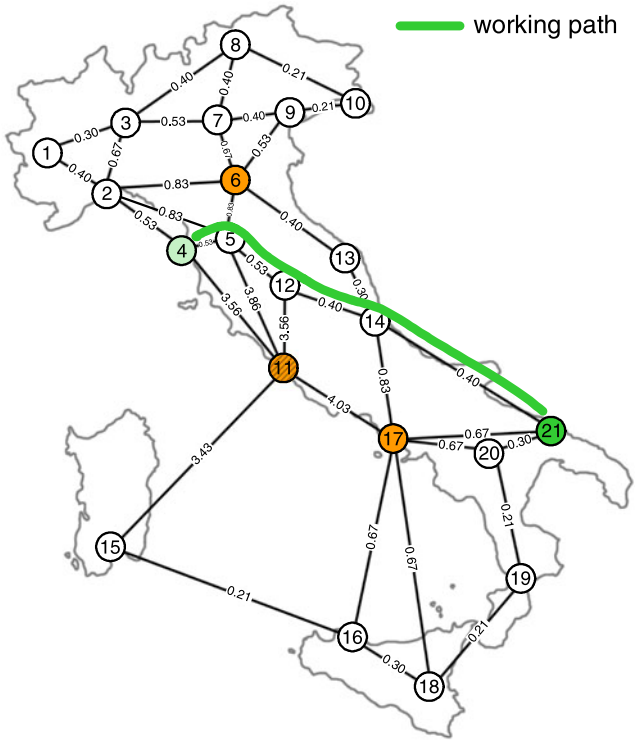


Fig. 11 Round 3 of the example execution of the proposed heuristic algorithm

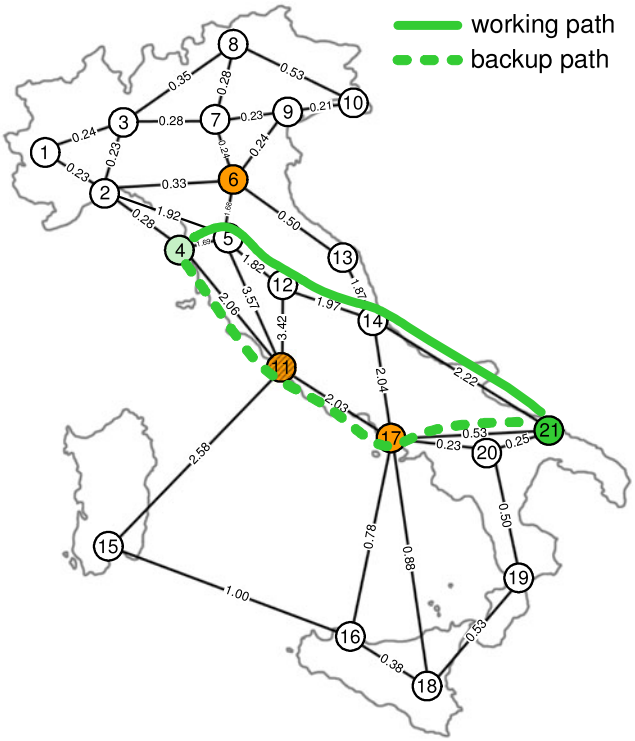


Fig. 12 Round 4 of the example execution of the proposed heuristic algorithm

– providing 100% of the requested capacity after a failure,
– protection against a failure of a single node,
– providing a single backup path to protect each working path (i.e. path protection scheme).

When routing the demands, a given incoming channel could be converted to any outgoing channel at a given transit node.

In our RA approach, working paths of anycast and unicast demands were calculated using the metric from (3) (that made them omit high-degree nodes, i.e. nodes being vulnerable to attacks). However, backup paths were found as the shortest ones using the standard metric of distance. Replica servers were located at nodes of the lowest degree (i.e. with the lowest possibility of being affected due to attacks). The properties of the RA approach were compared with the ones for the common NA technique of utilizing the standard distance metric in both working and backup path computations ((5) in both cases), and locating the anycast replica servers at high-degree nodes.

In both cases, we used the disjoint replica model implying that working and backup replica servers were located at different nodes.

Three scenarios of network load were investigated. In each case, a single set of anycast demands D_{AN} comprised all the network nodes. However, the analyzed sets of unicast demands consisted of the respective numbers of randomly chosen network node pairs implying three ratios of anycast demands ($|D_{AN}|/|D|$) equal to 10%, 20%, and 30% accordingly.

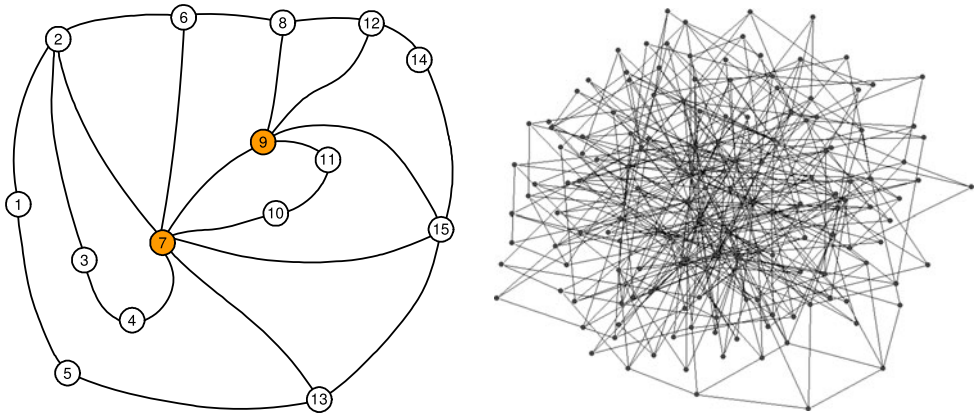
We analyzed three scenarios of the number replica servers (i.e. 2, 3 and 4) located at nodes of highest (NA model), or lowest (our RA model) degree. Indices of nodes implying the locations of anycast replica servers in the analyzed networks are given in Table 1.

A single simulation scenario was determined by the given network topology, the number of anycast replica servers, and the total number of demands $|D|$. For each scenario, computations were performed for 50 different sets of demands. When simulating attacks, the frequency of node failures was based on the values of $BC^*(n)$ coefficients.

5.1 Efficiency of the heuristic approach

In this section we evaluate the efficiency of the heuristic algorithm from Fig. 6 by comparing the results with the optimal ones achieved by finding the solution to the ILP problem defined by formulas (6)–(16). The comparison is presented in terms of the total link capacity necessary to establish the requested connections, the average length of the working and backup paths, as well as regarding the ratio of connection blocking. The results are given for two analyzed approaches, namely the introduced RA approach and the common NA technique. The results are given as a function of the varying network load (i.e. three ratios of anycast

MOST WIEDZY Downloaded from mostwiedzy.pl



(a) (b)

Fig. 13 Network topologies used in simulations: (**a**) ASF Network, (**b**) BA-150 Network

Table 1 Location of Anycast replica servers (node indices)

Network	Replica server location	2 replicas	3 replicas	4 replicas
ASF	Lowest degrees -RA	1, 3	1, 3, 4	1, 3, 4, 5
	Highest degrees -NA	7, 9	7, 9, 15	2, 7, 9, 15
BA-150	Lowest degrees -RA	20, 60	20, 60, 100	20, 60, 100, 137
	Highest degrees -NA	3, 5	3, 5, 10	3, 5, 10, 13

demands: $|D^{AN}|/|D|$ equal to 10%, 20%, and 30%, respectively), as well as the number of replica servers (i.e. 2, 3, 4, accordingly). For this set of experiments we assumed that all links offered $\Lambda = 40$ channels of equal (unitary) capacity.

Figure 14 shows the ratio of total link capacity utilization per connection as a function of the network load for the two analyzed approaches (NA and RA) using the ILP modeling and the heuristic algorithm. Table 2 presents the respective lengths of 95% confidence intervals. The results prove that the amount of capacity needed to establish the connections using the heuristic algorithm was comparable to the ILP results. In particular, as shown in Fig. 14, heuristic algorithm returned the results that were only about 3% worse on average, compared with the optimal ILP ones. For our RA approach, heuristic approach sometimes required less capacity (up to 2.49% less). However, such situation might occur, since the link cost of working paths given by formula (3) was not consistent with the hop metric.

In general, independent of the chosen way to find paths (ILP or heuristic), our RA approach required about 10% more capacity to establish survivable connections, compared to the common NA method.

Figure 15 shows the average ratio of total link capacity utilization per connection as a function of varying number of replica servers for the two analyzed methods (NA and RA). Table 3 shows the respective values of 95% confidence intervals. Results show that for each method, the analyzed average ratio remains practically at the same level, independent of the number of replica servers.

Figures 16–19 and Tables 4, 5 present the average lengths of working and backup paths obtained for the heuristic algorithm, compared with the optimal ILP ones. The first two figures (i.e. Figs. 16, 17) refer to the case of varying network load. Due to the inconsistency of the formula (3) used to find the working paths in the RA scenario with the hop metric, as well as owing to the difference between the objective function of the ILP model and the heuristic algorithm (to minimize the total cost of both connection paths, or the working path cost only, accordingly), the heuristic algorithm returned insignificantly shorter working paths compared to the ILP results (Fig. 16). Independent of the network load, for the NA approach the working paths were always shorter than the respective backup paths (which is a common case). However, in our RA method, owing to the requirement to omit high-degree nodes, working paths turned out to be longer than the associated backup paths.

Figures 18, 19 show the average lengths of paths as a function of varying number of replica servers for the two analyzed methods (NA and RA). Table 5 shows the respective values of 95% confidence intervals. Similar to the results of total link capacity utilization ratio per connection (Fig. 15), the analyzed values remain at the comparable level, independent of the number of replica servers.

MOST WIEDZY Downloaded from mostwiedzy.pl

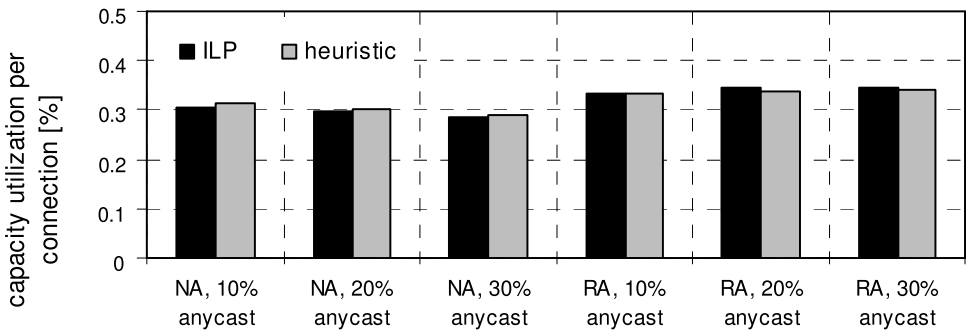


Fig. 14 Ratio of total link capacity utilization per connection obtained for the ILP model and using the heuristic algorithm for different network loads (number of replica servers: 2)

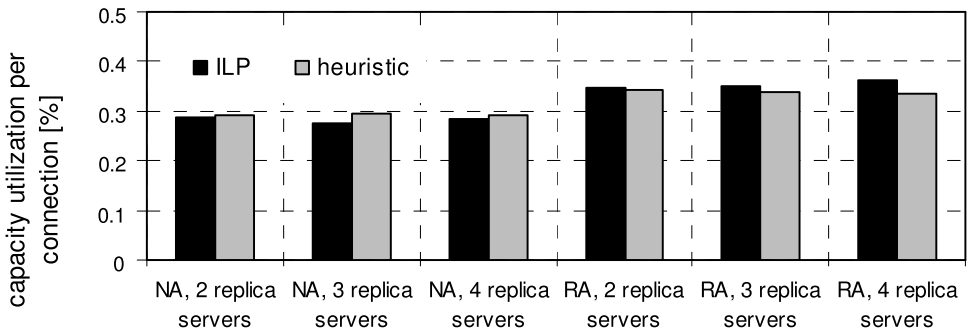


Fig. 15 Ratio of total link capacity utilization per connection for different numbers of replica servers (anycast ratio: 30%)

Table 2 Length of 95% confidence intervals for the average capacity utilization per connection (number of replica servers: 2) [%]

	NA, 10% anycast	NA, 20% anycast	NA, 30% anycast	RA, 10% anycast	RA, 20% anycast	RA, 30% anycast
ILP	0.000	0.006	0.005	0.000	0.007	0.006
heuristic	0.000	0.005	0.007	0.000	0.005	0.006

Table 3 Length of 95% confidence intervals for the average capacity utilization per connection (anycast ratio: 30%) [%]

	NA, 2 replica servers	NA, 3 replica servers	NA, 4 replica servers	RA, 2 replica servers	RA, 3 replica servers	RA, 4 replica servers
ILP	0.005	0.005	0.005	0.006	0.006	0.006
heuristic	0.006	0.006	0.006	0.006	0.006	0.006

Tables 6 and 7 present the average probability of demand blocking as a function of the network load, and the number of replica servers, accordingly. They show that ILP solver returned paths for all the demands. However, due to limitations on available link capacity, the demand blocking probability for the heuristic approach sometimes exceeded 0, but it was never greater than 0.034.

The remaining two parts of this section provide a more detailed evaluation of the proposed heuristic algorithm. In particular, apart from investigating the average length of working and backup paths, we present here also information referring to the average value of connection restoration time, as well as the total number of connections broken after attacks. Results are given for the two analyzed networks (ASF and BA-150), with the number of channels per link set to $\Lambda = 160$.

5.2 Detailed results for the heuristic algorithm (varying network load)

The number of available anycast replica servers is assumed here to be equal to 3. Figure 20 presents the average lengths of working and backup paths as a function of the network load obtained for the heuristic algorithm for the two inves-

MOST WIEDZY Downloaded from mostwiedzy.pl

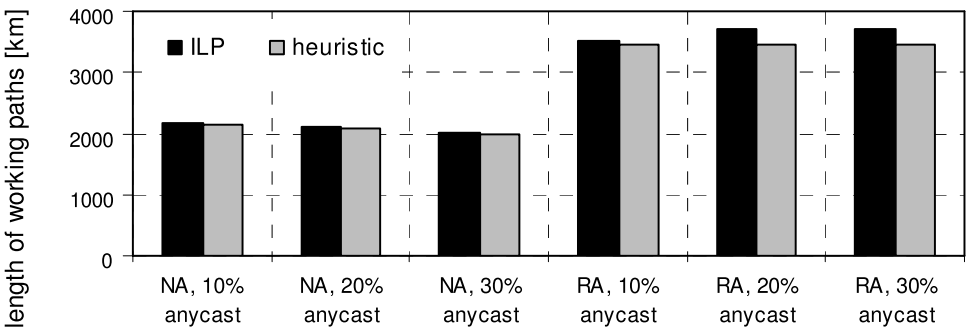


Fig. 16 Average length of working paths obtained for the heuristic algorithm and the ILP model for different network loads (number of replica servers: 2)

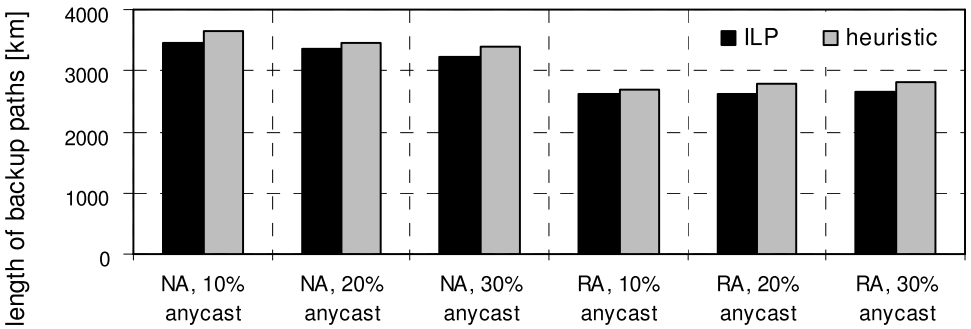


Fig. 17 Average length of backup paths obtained for the heuristic algorithm and the ILP model for different network loads (number of replica servers: 2)

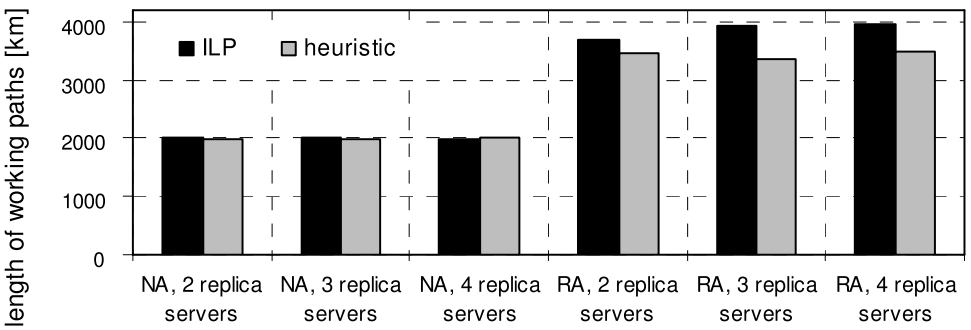


Fig. 18 Average length of working paths for different numbers of replica servers (anycast ratio: 30%)

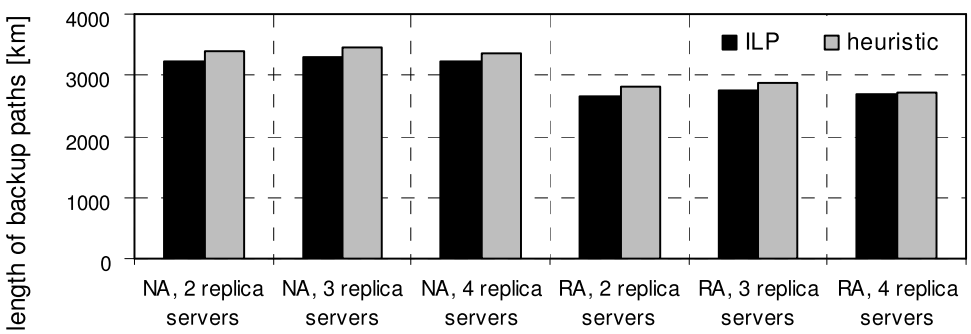


Fig. 19 Average length of backup paths for different numbers of replica servers (anycast ratio: 30%)

Table 4 Length of 95% confidence intervals for the average length of connection paths (number of replica servers: 2) [km]

	NA, 10% anycast	NA, 20% anycast	NA, 30% anycast	RA, 10% anycast	RA, 20% anycast	RA, 30% anycast
ILP, working paths	0.000	29.534	25.878	0.000	98.624	120.020
Heuristic, working paths	0.000	43.220	52.345	0.000	82.235	103.036
ILP, backup paths	0.000	29.534	25.877	0.000	43.382	56.783
Heuristic, backup paths	0.000	64.475	89.016	0.000	40.627	46.036

MOST WIEDZY Downloaded from mostwiedzy.pl

Table 5 Length of 95% confidence intervals for the average length of connection paths (anycast ratio: 30%) [km]

	NA, 2 replica servers	NA, 3 replica servers	NA, 4 replica servers	RA, 2 replica servers	RA, 3 replica servers	RA, 4 replica servers
ILP, working paths	26.114	31.465	28.958	120.888	126.096	129.329
Heuristic, working paths	52.345	53.687	53.688	103.036	105.612	108.320
ILP, backup paths	26.113	31.464	28.958	54.174	55.565	56.990
Heuristic, backup paths	89.016	90.298	91.298	46.036	47.186	48.396

Table 6 Demand blocking probability for varying network load (number of replica servers: 2)

	NA, 10% anycast	NA, 20% anycast	NA, 30% anycast	RA, 10% anycast	RA, 20% anycast	RA, 30% anycast
ILP	0.000	0.000	0.000	0.000	0.000	0.000
Heuristic	0.017	0.016	0.024	0.034	0.000	0.000

Table 7 Demand blocking probability for varying number of replica servers (anycast ratio: 30%)

	NA, 2 replica servers	NA, 3 replica servers	NA, 4 replica servers	RA, 2 replica servers	RA, 3 replica servers	RA, 4 replica servers
ILP	0.000	0.000	0.000	0.000	0.000	0.000
Heuristic	0.024	0.025	0.000	0.000	0.000	0.000

tigated approaches, i.e. NA and RA. Table 8 presents the respective lengths of 95% confidence intervals.

Results show that the obtained working paths for the proposed RA approach, are even up to 2.26 times longer, compared to the case of the common NA approach to find the shortest working paths using the formula (5). This was a direct implication of the fact that RA working paths tried to omit nodes of high-degree. However, for higher ratios of anycast demands ($|D_{AN}|/|D|$), this difference became smaller. In particular, it was greater for the BA-150 network. The reason for this is that BA-150 is a network obtained using more iterations of the Barabási-Albert algorithm of a scale-free network topology generation. Therefore this topology is "more scale-free" than the one of the 15-node ASF network.

Backup paths obtained using the proposed RA approach were up to 25% shorter, compared to the results for the common NA technique. This was a direct implication of the fact that only in the NA model, working paths were found as the shortest ones using the standard metrics of distance, which resulted in longer disjoint backup paths. Additionally, owing to the fact that NA working paths were established as the shortest ones and did not omit nodes of high degree, they were also shorter than the respective backup paths.

Figure 21 presents the total number of broken connections measured in the whole simulation as a function of the network load for the analyzed NA and RA approaches. Working paths in the RA model omitted high-degree nodes. As a result, up to 7.47 times less connections were broken after attacks for the RA approach, compared to the case of the NA technique. The difference was more visible for the larger network (BA-150).

Figure 22 shows the results referring to the average value of connection restoration time. Table 9 gives the respective lengths of 95% confidence intervals. In general the value of connection restoration time is mainly determined by the length of the backup path. Therefore, in our experiments the results for the proposed RA approach turned out to be about 17% better (i.e., smaller), compared to the respective ones for the common NA technique.

Table 10 presents the values of the demand blocking probability for the NA and RA approaches as a function of the network load. The number of available channels at each link (i.e. 160) was sufficient to establish all the connections. Therefore, the obtained values were equal to 0 in each case.

MOST WIEDZY Downloaded from mostwiedzy.pl

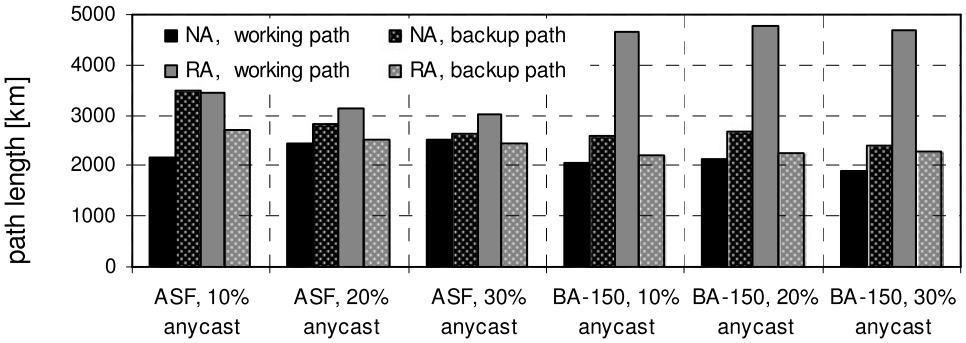


Fig. 20 Average length of working and backup paths

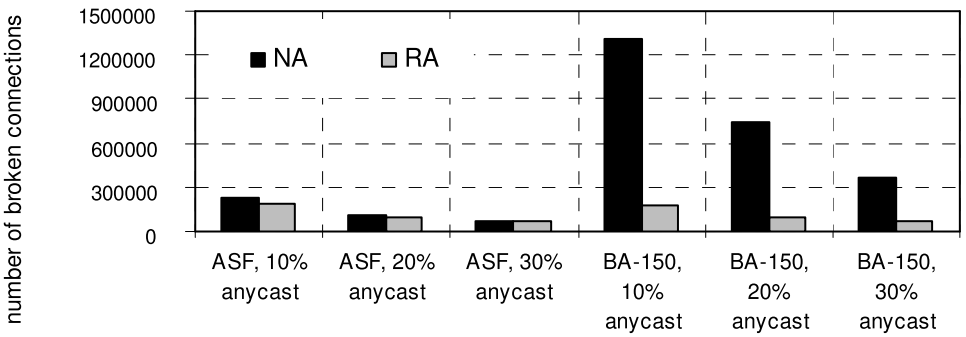


Fig. 21 Aggregate number of broken connections

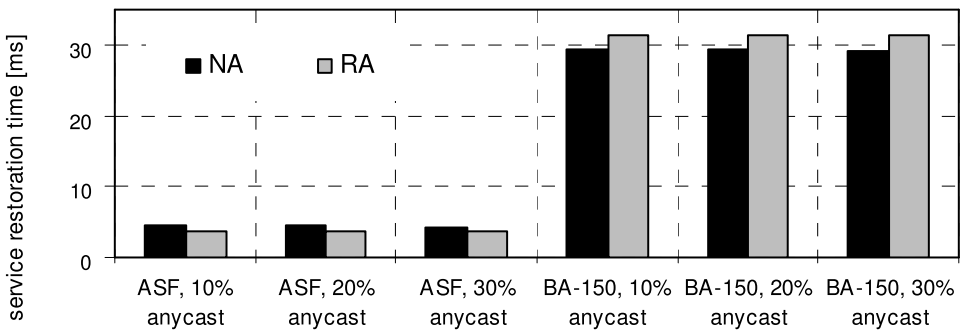


Fig. 22 Values of connection restoration time

Table 8 Length of 95% confidence intervals for the average length of connection paths [km]

	ASF, 10% anycast	ASF, 20% anycast	ASF, 30% anycast	BA-150, 10% anycast	BA-150, 20% anycast	BA-150, 30% anycast
NA, working path	0.000	43.220	52.345	21.713	23.330	25.436
NA, backup path	0.000	64.475	89.016	20.932	23.514	23.282
RA, working path	0.000	86.489	111.171	69.851	97.386	84.138
RA, backup path	0.000	42.728	49.670	19.013	20.301	22.768

Table 9 Length of 95% confidence intervals for the average value of connection restoration time [ms]

	ASF, 10% anycast	ASF, 20% anycast	ASF, 30% anycast	BA-150, 10% anycast	BA-150, 20% anycast	BA-150, 30% anycast
NA	0.521	1.115	1.525	0.518	0.366	0.291
RA	0.339	0.753	0.855	0.733	0.720	0.540

MOST WIEDZY Downloaded from mostwiedzy.pl

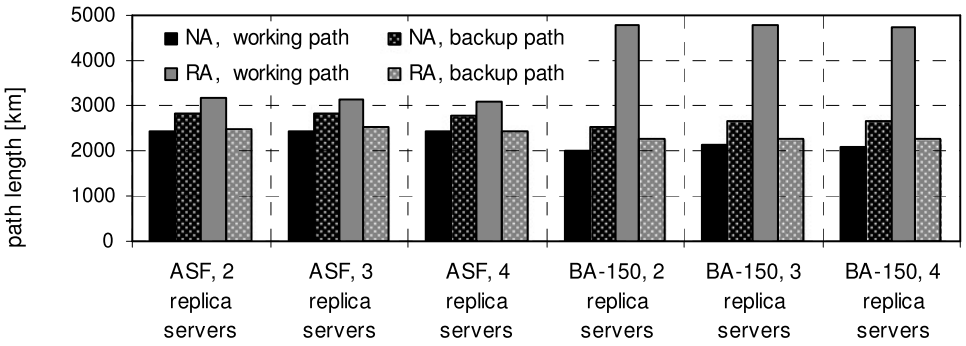


Fig. 23 Average length of working and backup paths

Fig. 24 Aggregate number of broken connections

Table 10 Demand blocking probability

	ASF, 10% anycast	ASF, 20% anycast	ASF, 30% anycast	BA-150, 10% anycast	BA-150, 20% anycast	BA-150, 30% anycast
NA	0.000	0.000	0.000	0.000	0.000	0.000
RA	0.000	0.000	0.000	0.000	0.000	0.000

Table 11 Length of 95% confidence intervals for the average length of working and backup paths [km]

	ASF, 2 replica servers	ASF, 3 replica servers	ASF, 4 replica servers	BA-150, 2 replica servers	BA-150, 3 replica servers	BA-150, 4 replica servers
NA, working path	42.512	43.220	44.710	23.330	23.330	23.330
NA, backup path	63.418	64.475	66.699	23.514	23.514	23.514
RA, working path	83.606	86.489	88.006	97.386	97.386	98.221
RA, backup path	41.304	42.728	43.478	20.301	20.301	20.541

5.3 Detailed results for the heuristic algorithm (varying number of replica servers)

The last set of experiments was to analyze the properties of the proposed heuristics for the case of varying number of anycast replica servers (i.e. 2, 3, and 4). The results are provided here for the case of the anycast demand ratio ($|D^{AN}|/|D|$) equal to 20%. Results presented in Figs. 23, 24, 25 and Tables 11, 12, 13 prove that the differences between all the analyzed characteristics regarding varying numbers of replica servers are negligible.

6 Conclusions

In this paper we discussed an important issue of protecting the communication networks against attacks. In particular,

MOST WIEDZY Downloaded from mostwiedzy.pl

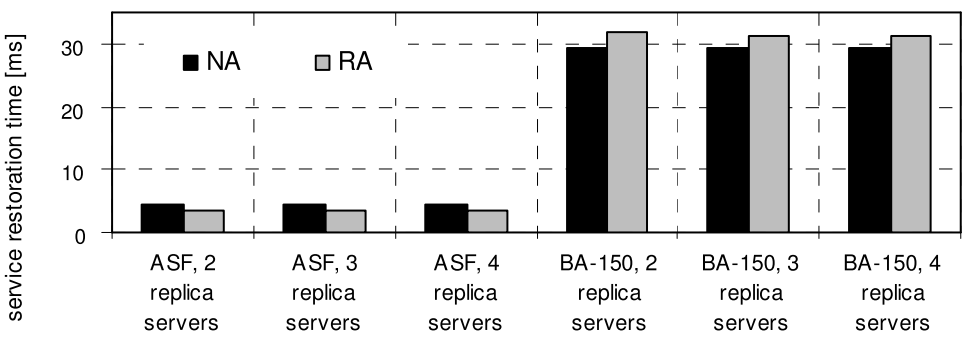


Fig. 25 Values of connection restoration time

Table 12 95% confidence intervals for the average value of connection restoration time [ms]

	ASF, 2 replica servers	ASF, 3 replica servers	ASF, 4 replica servers	BA-150, 2 replica servers	BA-150, 3 replica servers	BA-150, 4 replica servers
NA	0.111	0.112	0.115	0.035	0.037	0.041
RA	0.067	0.075	0.072	0.058	0.072	0.083

Table 13 Demand blocking probability

	ASF, 2 replica servers	ASF, 3 replica servers	ASF, 4 replica servers	BA-150, 2 replica servers	BA-150, 3 replica servers	BA-150, 4 replica servers
NA	0.000	0.000	0.000	0.000	0.000	0.000
RA	0.000	0.000	0.000	0.000	0.000	0.000

we focused on joint protection of anycast and unicast flows. The main achievement of the paper was a new approach (which we called RA) to provide protection of anycast and unicast flows against attacks, dedicated to networks of irregular topology.

The main features included the introduction of a special metric to be used in working path computations to make these paths omit high-degree nodes, and locating the anycast replica servers at low-degree nodes.

The respective ILP model was proposed and followed by a time-efficient heuristic algorithm. The efficiency of the heuristic algorithm was shown based on the results of simulation experiments. The most significant achievement of our RA approach was the remarkable decrease (up to 7.47 times) of the number of connections broken after attacks, compared to the results for the common NA case of using the standard distance metric to find both the working and backup paths, as well as locating the replica servers at high-degree nodes.

Acknowledgements The work of J. Rak was supported by the Polish Ministry of Science and Higher Education under the grant N N519 581038. The work of K. Walkowiak was supported by the Polish Ministry of Science and Higher Education under the grant N N516 070435.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Ali, M. (2004). Shareability in optical networks: beyond bandwidth optimization. *IEEE Optical Communications*, *42*(2), s11–s15.
2. Autenrieth, A. (2003). Recovery time analysis of differentiated resilience in MPLS. In *Proceedings of design of reliable communication networks'03-DRCN'03* (pp. 333–340).
3. Autenrieth, A., & Kirstaedler, A. (2002). Engineering end-to-end resilience using resilience differentiated QoS. *IEEE Communications Magazine*, *40*(1), 50–57.
4. Awerbuch, B., Brinkmann, A., & Scheideler, C. (2003). Anycasting adversarial systems: routing and admission control. In *LNCS: Vol. 2719* (pp. 1153–1168). Berlin: Springer.
5. Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*, 509–511.
6. Barr, R. S., & Kingsley, M. S. (2003). *Grooming telecommunications networks: optimization models and methods* (Technical report-EMIS-03). Southern Methodist University, Dallas, USA, 1–27.
7. Buford, J., Yu, H., & Lua, E. (2009). *P2P networking and applications*. San Mateo: Morgan Kaufmann.

MOST WIEDZY Downloaded from mostwiedzy.pl

8. Chow, T. Y., Chudak, F., & Ffrench, A. M. (2004). Fast optical layer mesh protection using pre-cross connected trails. *IEEE/ACM Transactions on Networking*, *12*(3), 539–548.
9. Dijkstra, E. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, *1*, 269–271.
10. Goh, K.-I., Oh, E. S., Jeong, H., Kahng, B., & Kim, D. (2002). Classification of scale free networks. arXiv:cond-mat/0205232, v2 10 Oct. 2002.
11. Grover, W. D., & Stamatelakis, D. (1998). Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration. In *Proc. IEEE ICC'98* (Vol. 1, pp. 537–543).
12. Ho, P.-H., & Mouftah, H. T. (2004). Shared protection in mesh WDM. *Networks*, *42*(1), 70–76.
13. Hofmann, M., & Beaumont, L. (2005). *Content networking: architecture, protocols, and practice*. San Mateo: Morgan Kaufmann.
14. Hyytia, E. (2004). Heuristic algorithms for the generalized routing and wavelength assignment problem. In *Proceedings of 17th Nordic teletraffic seminar NTS-17* (pp. 373–386).
15. ILOG AMPL/CPLEX software: www.ilog.com/products/cplex/.
16. Jaumard, B., & Sebbah, S. (2009). PWCE design in survivable WDM networks using unrestricted shape *p*-structure patterns. In *Proc. IEEE Sarnoff'09* (pp. 1–5).
17. Ma, H., Fayek, D., & Ho, P.-H. (2008). Availability-constrained multipath protection in backbone networks with double-link failure. In *Proc. IEEE ICC'08* (pp. 158–164).
18. Mukherjee, B. (2006). *Optical WDM networks*. Berlin: Springer.
19. Rak, J. (2010). *k*-Penalty: a novel approach to find *k*-disjoint paths with differentiated path costs. *IEEE Communications Letters*, *14*(4), 354–356.
20. Ramasubramanian, S., & Chandak, A. (2008). Dual-link failure resiliency through backup link mutual exclusion. *IEEE/ACM Transactions on Networking*, *16*(1), 157–169.
21. Vasseur, J.-P., Pickavet, M., & Demeester, P. (2004). *Network recovery*. San Mateo: Morgan Kaufmann.
22. Walkowiak, K. (2007). *Anycast communication—a new approach to survivability of connection-oriented networks, CCIS, 1* (pp. 378–389) Berlin: Springer.
23. Zhou, S., & Mondragón, R. J. (2004). The rich-club phenomenon in the Internet topology. *IEEE Communications Letters*, *8*(3), 180–182.

Jacek Rak holds M.Sc. and Ph.D. degrees in computer science (options: computer networks, and computer communications, accordingly) received with distinction in 2003 and 2009, respectively from Gdansk University of Technology (GUT), Poland. He is currently an assistant professor at the Department of Computer Communications of the Faculty of Electronics, Telecommunications and Informatics at GUT. His main research areas include: routing, design, dimensioning and analysis of high-speed wavelength routed backbone networks with focus on survivability.

Dr. Rak was involved in many projects related to optimization of reliable computer networks. He was the TPC Co-Chair of NETWORKS 2010 Conference, Publication Chair of NETWORKS 2010 and BCFIC 2011 conferences. He also served as a TPC member/reviewer of important conferences on communications, e.g. IEEE ICC, IEEE GLOBECOM, DRCN, and journals, e.g. IEEE/ACM Transactions on Networking, or IEEE Communications Letters.

Dr. Rak is a member of IEEE (and IEEE Communications Society), IFIP TC6 WG 6.10 (Photonic Networking Group), as well as Polish Information Society (PTI). He is also the founder and the General Chair of the International Workshop on Reliable Networks Design and Modeling (RNDM).

Krzysztof Walkowiak was born in 1973. He received the Ph.D. degree and the D.Sc. (habilitation) degree in computer science from the Wroclaw University of Technology, Poland, in 2000 and 2008, respectively. Currently, he is an Associate Professor at the Chair of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Technology. His research interest is mainly focused on optimization of network distributed systems like P2P systems, multicasting systems, Grid systems; network survivability; optimization of connection-oriented networks (MPLS, DWDM); application of soft-optimization techniques for design of computer networks. Prof. Walkowiak was involved in many research projects related to optimization of computer networks. Moreover, he was consulting projects for the large companies including TP SA, PZU, PKO BP, Energia Pro, Ernst and Young. Prof. Walkowiak published more than 120 scientific papers. He serves as a reviewer for many international journals including: Computer Communications, Future Generation Computer Systems, Computational Optimization and Applications, International Journal of Applied Mathematics and Computer Science, Expert Systems, Pattern Analysis and Applications, International Journal of Computer Mathematics. He is/was actively involved in many international conferences. Prof. Walkowiak is a member of IEEE and ComSoc.

MOST WIEDZY Downloaded from mostwiedzy.pl